**LAPORAN TUGAS ALGORITMA DAN STRUKTUR DATA**

**JOBSHEET 6**

`

NAMA : RADITYA RIEFKI

KELAS : TI 1E

ABSEN : 23

**6.6 Praktikum 1 - Mengimplementasikan Sorting menggunakan object**

a. SORTING – BUBBLE SORT

1. Kode Program

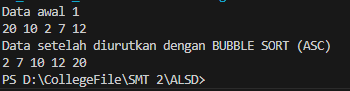
Sorting

|  |
| --- |
| public class Sorting23 {  int[] data;  int jumData;  Sorting23 (int Data[], int jmlData){  jumData = jmlData;  data = new int [jmlData];  for (int i = 0; i < jumData; i++) {  data[i] = Data[i];  }  }  void bubbleSort(){  int temp = 0;  for (int i = 0; i < jumData - 1; i++) {  for (int j = 1; j < jumData - i; j++) {  if (data[j-1] > data[j]) {  temp = data[j];  data[j] = data[j-1];  data[j-1] = temp;    }  }    }  }  void tampil(){  for (int i = 0; i < jumData; i++) {  System.out.print(data[i] + " ");  }  System.out.println();  }  } |

Sorting Main

|  |
| --- |
| public class SortingMain23 {  public static void main(String[] args) {  int a [] = {20, 10, 2, 7, 12};  Sorting23 dataurut1 = new Sorting23(a, a.length);  System.out.println("Data awal 1");  dataurut1.tampil();  dataurut1.bubbleSort();  System.out.println("Data setelah diurutkan dengan BUBBLE SORT (ASC)");  dataurut1.tampil();    }  } |

OUTPUT



b. SORTING – SELECTION SORT

1. Kode Program

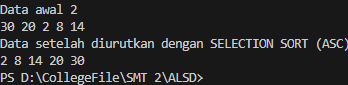
Sorting

|  |
| --- |
| public class Sorting23 {  int[] data;  int jumData;  Sorting23 (int Data[], int jmlData){  jumData = jmlData;  data = new int [jmlData];  for (int i = 0; i < jumData; i++) {  data[i] = Data[i];  }  }  void bubbleSort(){  int temp = 0;  for (int i = 0; i < jumData - 1; i++) {  for (int j = 1; j < jumData - i; j++) {  if (data[j-1] > data[j]) {  temp = data[j];  data[j] = data[j-1];  data[j-1] = temp;    }  }    }  }  void tampil(){  for (int i = 0; i < jumData; i++) {  System.out.print(data[i] + " ");  }  System.out.println();  }  void SelectionSort(){  for (int i = 0; i < jumData - 1; i++) {  int min = i;  for (int j = i + 1; j < jumData; j++) {  if (data[j] < data[min]) {  min = j;  }  int temp = data[i];  data[i] = data[min];  data[min] = temp;    }    }  }  } |

Sorting Main

|  |
| --- |
| public class SortingMain23 {  public static void main(String[] args) {  int a [] = {20, 10, 2, 7, 12};  int b[] = {30, 20, 2, 8, 14};  Sorting23 dataurut1 = new Sorting23(a, a.length);  Sorting23 dataurut2 = new Sorting23(a, b.length);  System.out.println("Data awal 1");  dataurut1.tampil();  dataurut1.bubbleSort();  System.out.println("Data setelah diurutkan dengan BUBBLE SORT (ASC)");  dataurut1.tampil();    System.out.println("Data awal 2");  dataurut2.tampil();  dataurut2.selectionSort();  System.out.println("Data setelah diurutkan dengan SELECTION SORT (ASC)");  dataurut2.tampil();    }  } |

2. Output



c. SORTING – INSERTION SORT

1. Kode Program

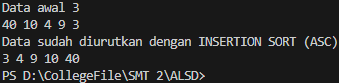
Sorting

|  |
| --- |
| public class Sorting23 {  int[] data;  int jumData;  Sorting23 (int Data[], int jmlData){  jumData = jmlData;  data = new int [jmlData];  for (int i = 0; i < jumData; i++) {  data[i] = Data[i];  }  }  void bubbleSort(){  int temp = 0;  for (int i = 0; i < jumData - 1; i++) {  for (int j = 1; j < jumData - i; j++) {  if (data[j-1] > data[j]) {  temp = data[j];  data[j] = data[j-1];  data[j-1] = temp;    }  }    }  }  void tampil(){  for (int i = 0; i < jumData; i++) {  System.out.print(data[i] + " ");  }  System.out.println();  }  void SelectionSort(){  for (int i = 0; i < jumData - 1; i++) {  int min = i;  for (int j = i + 1; j < jumData; j++) {  if (data[j] < data[min]) {  min = j;  }  int temp = data[i];  data[i] = data[min];  data[min] = temp;    }    }  }  void insertionSort(){  for (int i = 1; i <= data.length - 1; i++) {  int temp = data[i];  int j = i - 1;  while (j >= 0 && data[j] > temp) {  data[j + 1] = data [j];  j --;  }  data[j + 1] = temp;  }  }  } |

Sorting Main

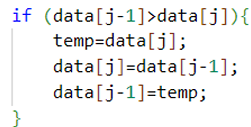
|  |
| --- |
| public class SortingMain23 {  public static void main(String[] args) {  int a [] = {20, 10, 2, 7, 12};  int b [] = {30, 20, 2, 8, 14};  int c [] = {40, 10, 4, 9 , 3};  Sorting23 dataurut1 = new Sorting23(a, a.length);  Sorting23 dataurut2 = new Sorting23(b, b.length);  Sorting23 dataurut3 = new Sorting23(c, c.length);  System.out.println("Data awal 1");  dataurut1.tampil();  dataurut1.bubbleSort();  System.out.println("Data setelah diurutkan dengan BUBBLE SORT (ASC)");  dataurut1.tampil();    System.out.println("Data awal 2");  dataurut2.tampil();  dataurut2.SelectionSort();  System.out.println("Data setelah diurutkan dengan SELECTION SORT (ASC)");  dataurut2.tampil();  System.out.println("Data awal 3");  dataurut3.tampil();  dataurut3.insertionSort();  System.out.println("Data sudah diurutkan dengan INSERTION SORT (ASC)");  dataurut3.tampil();    }  } |

2. Output



**6.2.5 Pertanyaan!**

1. Jelaskan fungsi kode program berikut



- Pada line pertama mengecek elemen sebelumnya data[ j – 1] apakah lebih besar dari elemen sekarang data [j]

Jika true nilai disimpan di variabel temp

temp = data[j] nilai dari temp disimpan ke data [j]

data [j] = data[ j – 1 ] geser data[j -1] ke data[j]

data[j – 1] = temp memindah nilai asli data[j] ke data [j-1]

maka elemen yang lebih kecil akan di swap ke posisi yang lebih depan dalam array

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!

- for (int j = i + 1; j < jumData; j++) dan

if (data[j] < data[min]) { min = j; }

3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan



Menggeser elemen yang lebih besar ke kanan agar elemen terkecil yang sedang di insert diletakkan pada posisi paling depan

4. Pada Insertion sort, apakah tujuan dari perintah



Menggeser elemen ke kanan agar elemen terkecil berada di posisi terdepan

**6.7 Praktikum 2- (Sorting Menggunakan Array of Object)**

6.3.1 Langkah Praktikum 2 - Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

1. Kode Program

Mahasiswa

|  |
| --- |
| public class Mahasiswa23 {  String nim;  String nama;  String kelas;  double ipk;  Mahasiswa23(){  }  Mahasiswa23(String nm, String name, String kls, double ip){  nim = nm;  nama = name;  ipk = ip;  kelas = kls;  }  void tampilInformasi(){  System.out.println("Nama: " + nama);  System.out.println("NIM: " + nim);  System.out.println("IPK: " +ipk);  System.out.println("Kelas: " + kelas);  }  } |

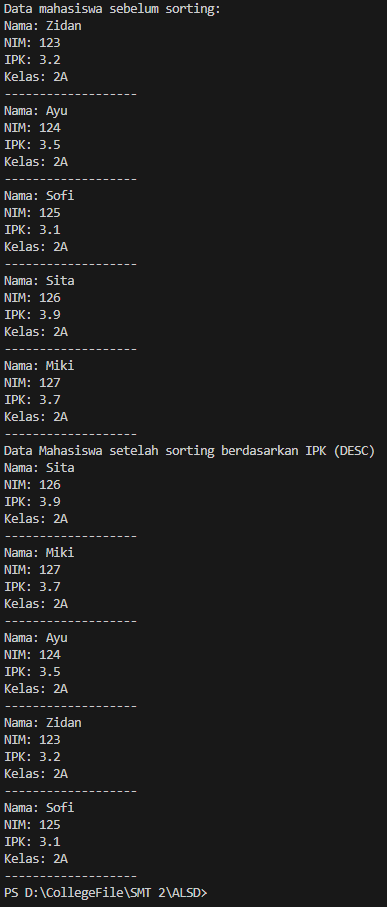
Mahasiswa Berprestasi

|  |
| --- |
| public class MahasiswaBerprestasi23 {  Mahasiswa23[] listMhs = new Mahasiswa23 [5];  int idx;  void tambah (Mahasiswa23 m){  if (idx < listMhs.length) {  listMhs[idx] = m;  idx++;  }else {  System.out.println("Data sudah penuh");  }  }  void tampil(){  for (Mahasiswa23 m : listMhs) {  m.tampilInformasi();  System.out.println("-------------------");  }  }  void bubbleSort(){  for (int i = 0; i < listMhs.length-1; i++){  for (int j = 1; j < listMhs.length - i; j++){  if (listMhs[j].ipk > listMhs[j-1].ipk) {  Mahasiswa23 tmp = listMhs[j];  listMhs[j] = listMhs[j-1];  listMhs[j-1] = tmp;    }  }  }  }  } |

Mahasiswa Demo

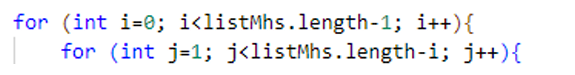
|  |
| --- |
| public class MahasiswaDemo23 {  public static void main(String[] args) {  MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23();  Mahasiswa23 m1 = new Mahasiswa23("123", "Zidan", "2A", 3.2);  Mahasiswa23 m2 = new Mahasiswa23("124", "Ayu", "2A", 3.5);  Mahasiswa23 m3 = new Mahasiswa23("125", "Sofi", "2A", 3.1);  Mahasiswa23 m4 = new Mahasiswa23("126", "Sita", "2A", 3.9);  Mahasiswa23 m5 = new Mahasiswa23("127", "Miki", "2A", 3.7);  list.tambah(m1);  list.tambah(m2);  list.tambah(m3);  list.tambah(m4);  list.tambah(m5);  System.out.println("Data mahasiswa sebelum sorting: ");  list.tampil();  System.out.println("Data Mahasiswa setelah sorting berdasarkan IPK (DESC)");  list.bubbleSort();  list.tampil();  }  } |

2. Output



**6.3.4 Pertanyaan**

1**.** Perhatikan perulangan di dalam bubbleSort() di bawah ini:

****

a. Mengapa syarat dari perulangan i adalah i

- untuk menghindari iterasi yang tidak diperlukan

b. Mengapa syarat dari perulangan j adalah j < listMhs.length-i?

- untuk menghindari akses index yang melebihi batas

c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

- perulangan dilakukan 49 kali karena i<listMhs.length-1 dan 49 tahap sorting

2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyborad) yang terdiri dari nim, nama, kelas, dan ipk!

Kode program MahasiswaBerprestasi23

|  |
| --- |
| public class MahasiswaBerprestasi23 {  Mahasiswa23[] listMhs;  int idx;  MahasiswaBerprestasi23(int jumlah){  listMhs = new Mahasiswa23[jumlah];  idx = 0;  }  void tambah (Mahasiswa23 m){  if (idx < listMhs.length) {  listMhs[idx] = m;  idx++;  }else {  System.out.println("Data sudah penuh");  }  }  void tampil() {  for (int i = 0; i < idx; i++) {  listMhs[i].tampilInformasi();  }  }  void bubbleSort(){  for (int i = 0; i < listMhs.length-1; i++){  for (int j = 1; j < listMhs.length - i; j++){  if (listMhs[j].ipk > listMhs[j-1].ipk) {  Mahasiswa23 tmp = listMhs[j];  listMhs[j] = listMhs[j-1];  listMhs[j-1] = tmp;    }  }  }  }  } |

Kode Program MahasiswaDemo

|  |
| --- |
| import java.util.Scanner;  public class MahasiswaDemo23 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);    System.out.print("Masukkan jumlah mahasiswa: ");  int jumlah = sc.nextInt();  sc.nextLine();    MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23(jumlah);    for (int i = 0; i < jumlah; i++) {  System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");  System.out.print("Nama: ");  String nama = sc.nextLine();  System.out.print("NIM: ");  String nim = sc.nextLine();  System.out.print("IPK: ");  double ipk = sc.nextDouble();  sc.nextLine();  System.out.print("Kelas: ");  String kelas = sc.nextLine();  Mahasiswa23 m = new Mahasiswa23(nim, nama, kelas, ipk);  list.tambah(m);  }  System.out.println("\nData mahasiswa sebelum sorting:");  list.tampil();  System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC):");  list.bubbleSort();  list.tampil();  }  } |

**6.4 Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)**

6.4.1. Langkah-langkah Percobaan.

1. Kode Program

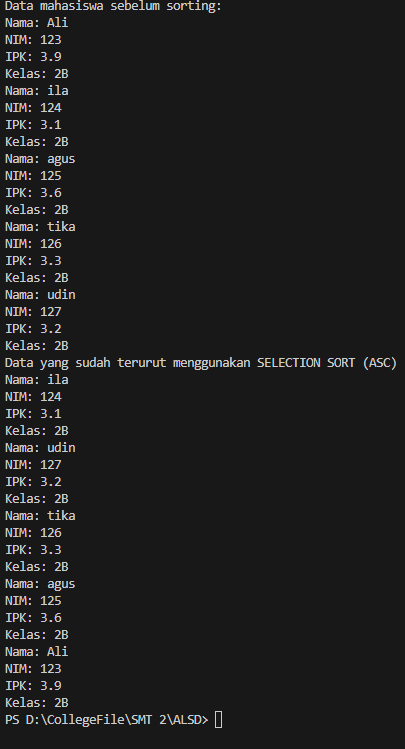
Mahasiswa Berprestasi

|  |
| --- |
| public class MahasiswaBerprestasi23 {  Mahasiswa23[] listMhs;  int idx;  MahasiswaBerprestasi23(int jumlah){  listMhs = new Mahasiswa23[jumlah];  idx = 0;  }  void tambah (Mahasiswa23 m){  if (idx < listMhs.length) {  listMhs[idx] = m;  idx++;  }else {  System.out.println("Data sudah penuh");  }  }  void tampil() {  for (int i = 0; i < idx; i++) {  listMhs[i].tampilInformasi();  }  }  void bubbleSort(){  for (int i = 0; i < listMhs.length-1; i++){  for (int j = 1; j < listMhs.length - i; j++){  if (listMhs[j].ipk > listMhs[j-1].ipk) {  Mahasiswa23 tmp = listMhs[j];  listMhs[j] = listMhs[j-1];  listMhs[j-1] = tmp;    }  }  }  }  void selectionSort(){  for (int i = 0; i < listMhs.length; i++) {  int idxMin = i;  for (int j = i+1; j < listMhs.length; j++) {  if (listMhs[j].ipk < listMhs[idxMin].ipk){  idxMin = j;  }  }  Mahasiswa23 tmp = listMhs[idxMin];  listMhs[idxMin] = listMhs[i];  listMhs[i] = tmp;  }  }  } |

Mahasiswa Demo

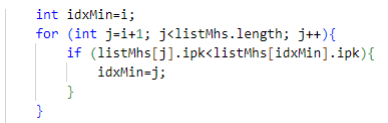
|  |
| --- |
| import java.util.Scanner;  public class MahasiswaDemo23 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);    System.out.print("Masukkan jumlah mahasiswa: ");  int jumlah = sc.nextInt();  sc.nextLine();    MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23(jumlah);    for (int i = 0; i < jumlah; i++) {  System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");  System.out.print("NIM: ");  String nim = sc.nextLine();  System.out.print("Nama: ");  String nama = sc.nextLine();  System.out.print("IPK: ");  double ipk = sc.nextDouble();  sc.nextLine();  System.out.print("Kelas: ");  String kelas = sc.nextLine();  Mahasiswa23 m = new Mahasiswa23(nim, nama, kelas, ipk);  list.tambah(m);  }  System.out.println("\nData mahasiswa sebelum sorting:");  list.tampil();  System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC):");  list.bubbleSort();  list.tampil();    System.out.println("Data yang sudah terurut menggunakan SELECTION SORT (ASC)");  list.selectionSort();  list.tampil();  }  } |

2. Output



**6.4.3 Pertanyaan**

Di dalam method selection sort, terdapat baris program seperti di bawah ini:

****

Pada line pertama untuk menyimpan index elemen minimum

Pada line ke 2 melakukan iterasi dari elemen setelah I hingga akhir array untuk mencari elemen array denga IPK terkecil

Pada line ke 3 jika menemukan elemen ipk terkecil maka disimpan ke idx min dengan indeks elemen tersebut

Pada line ke 4 saat kondisi if terpenuhi menyimpan nilai idxmin ke variable j

**6.5 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort**

6.5.1 Langkah-langkah Percobaan

1. Kode Program

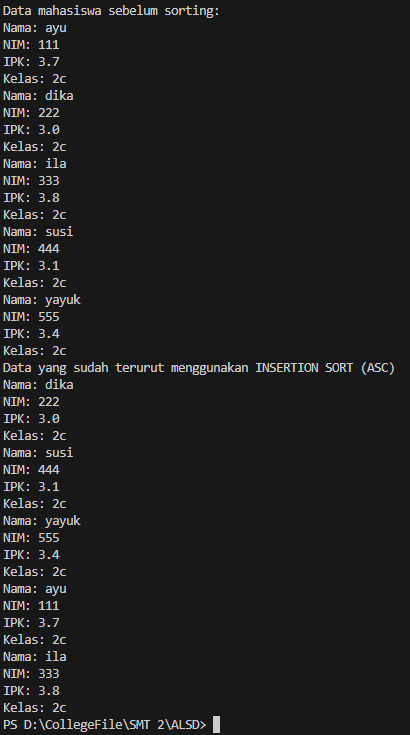
Mahasiswa Berprestasi

|  |
| --- |
| public class MahasiswaBerprestasi23 {  Mahasiswa23[] listMhs;  int idx;  MahasiswaBerprestasi23(int jumlah){  listMhs = new Mahasiswa23[jumlah];  idx = 0;  }  void tambah (Mahasiswa23 m){  if (idx < listMhs.length) {  listMhs[idx] = m;  idx++;  }else {  System.out.println("Data sudah penuh");  }  }  void tampil() {  for (int i = 0; i < idx; i++) {  listMhs[i].tampilInformasi();  }  }  void bubbleSort(){  for (int i = 0; i < listMhs.length-1; i++){  for (int j = 1; j < listMhs.length - i; j++){  if (listMhs[j].ipk > listMhs[j-1].ipk) {  Mahasiswa23 tmp = listMhs[j];  listMhs[j] = listMhs[j-1];  listMhs[j-1] = tmp;    }  }  }  }  void selectionSort(){  for (int i = 0; i < listMhs.length; i++) {  int idxMin = i;  for (int j = i+1; j < listMhs.length; j++) {  if (listMhs[j].ipk < listMhs[idxMin].ipk){  idxMin = j;  }  }  Mahasiswa23 tmp = listMhs[idxMin];  listMhs[idxMin] = listMhs[i];  listMhs[i] = tmp;  }  }  void insertionSort(){  for (int i = 1; i < listMhs.length; i++) {  Mahasiswa23 temp = listMhs[i];  int j = i;  while (j > 0 && listMhs[j-1].ipk > temp.ipk){  listMhs[j] = listMhs[j-1];  j--;  }  listMhs[j] = temp;  }  }  } |

Mahasiswa Demo

|  |
| --- |
| import java.util.Scanner;  public class MahasiswaDemo23 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);    System.out.print("Masukkan jumlah mahasiswa: ");  int jumlah = sc.nextInt();  sc.nextLine();    MahasiswaBerprestasi23 list = new MahasiswaBerprestasi23(jumlah);    for (int i = 0; i < jumlah; i++) {  System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");  System.out.print("NIM: ");  String nim = sc.nextLine();  System.out.print("Nama: ");  String nama = sc.nextLine();  System.out.print("Kelas: ");  String kelas = sc.nextLine();  System.out.print("IPK: ");  double ipk = sc.nextDouble();  sc.nextLine();  Mahasiswa23 m = new Mahasiswa23(nim, nama, kelas, ipk);  list.tambah(m);  }  System.out.println("\nData mahasiswa sebelum sorting:");  list.tampil();  System.out.println("Data yang sudah terurut menggunakan INSERTION SORT (ASC)");  list.insertionSort();  list.tampil();  }  } |

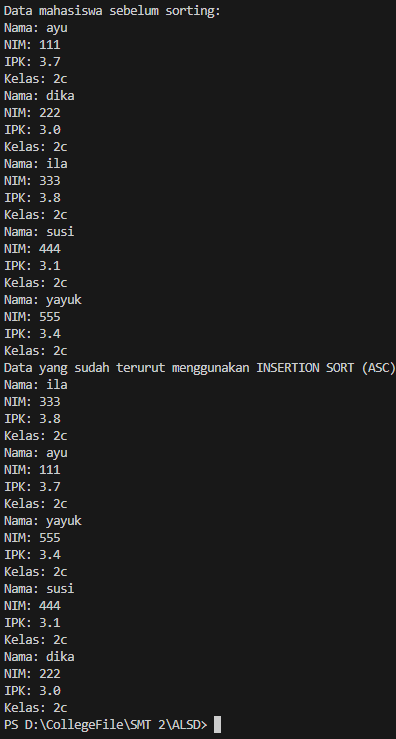
2. Output



**6.5.3 Pertanyaan**

Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending

|  |
| --- |
| **void insertionSort(){**  **for (int i = 1; i < listMhs.length; i++) {**  **Mahasiswa23 temp = listMhs[i];**  **int j = i;**  **while (j > 0 && listMhs[j-1].ipk < temp.ipk){**  **listMhs[j] = listMhs[j-1];**  **j--;**  **}**  **listMhs[j] = temp;**  **}**  **}** |



**6.6 Latihan Praktikum**

1.Kode Program

Dosen

|  |
| --- |
| public class Dosen23 {  String kode;  String nama;  Boolean jenisKelamin;  int usia;  int i = 0;  Dosen23(){  }  Dosen23(String kd, String name,Boolean jk,int age){  kode = kd;  nama = name;  jenisKelamin = jk;  usia = age;  }  void tampil(){  System.out.println("Data Dosen ke- " + (i+1));  System.out.println("Kode: " +kode);  System.out.println("Nama: " +nama);  if(jenisKelamin==true){  System.out.println("Jenis Kelamin: Perempuan");  }else{  System.out.println("Jenis Kelamin: Laki-laki");  }  System.out.println("Usia: " +usia);  System.out.println();    }  } |

Data Dosen

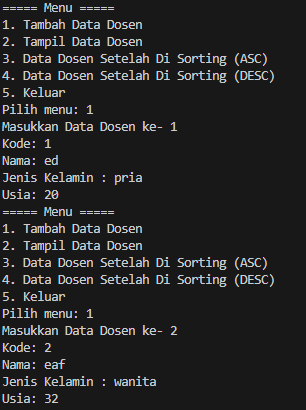
|  |
| --- |
| public class DataDosen23 {  Dosen23 [] arrayDosen23 = new Dosen23[10];  int idx;  void tambah(Dosen23 dsn){  if (idx < arrayDosen23.length) {  arrayDosen23[idx] = dsn;  idx++;  } else {  System.out.println("Data sudah penuh");  }  }    void tampil() {  for (int i = 0; i < idx; i++) {  arrayDosen23[i].tampil();  }  }  void sortingAsc() {  for (int i = 0; i < idx - 1; i++) {  for (int j = 0; j < idx - 1 - i; j++) {  if (arrayDosen23[j].usia > arrayDosen23[j + 1].usia) {  Dosen23 temp = arrayDosen23[j];  arrayDosen23[j] = arrayDosen23[j + 1];  arrayDosen23[j + 1] = temp;  }  }  }  }  void sortingDesc() {  for (int i = 1; i < idx; i++) {  Dosen23 temp = arrayDosen23[i];  int j = i;  while (j > 0 && arrayDosen23[j - 1].usia < temp.usia) {  arrayDosen23[j] = arrayDosen23[j - 1];  j--;  }  arrayDosen23[j] = temp;  }  }    } |

Dosen Main

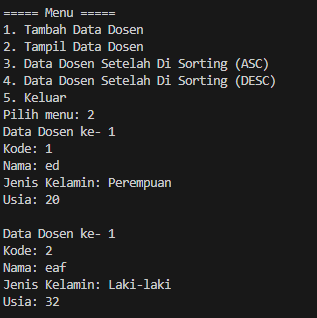
|  |
| --- |
| import java.util.Scanner;  public class DosenMain23 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  DataDosen23 dosen = new DataDosen23();  String kode, nama;  Boolean jk;  int usia;  int i = 0;  while (true) {  System.out.println("===== Menu =====");  System.out.println("1. Tambah Data Dosen");  System.out.println("2. Tampil Data Dosen");  System.out.println("3. Data Dosen Setelah Di Sorting (ASC)");  System.out.println("4. Data Dosen Setelah Di Sorting (DESC)");  System.out.println("5. Keluar");  System.out.print("Pilih menu: ");  int menu = sc.nextInt();  sc.nextLine();  switch (menu) {  case 1:  System.out.println("Masukkan Data Dosen ke- " +(i+1));  System.out.print("Kode: ");  kode = sc.nextLine();  System.out.print("Nama: ");  nama = sc.nextLine();  while (true) {  System.out.print("Jenis Kelamin : ");  String dummy = sc.nextLine();  jk = Boolean.parseBoolean(dummy);    if (dummy.equalsIgnoreCase("Pria")) {  jk = true;  break;  } else if (dummy.equalsIgnoreCase("Wanita")) {  jk = false;  break;  }else  System.out.println("Pilihan tidak valid, silahkan memilih Pria/Wanita");  }    System.out.print("Usia: ");  usia = sc.nextInt();  sc.nextLine();  dosen.tambah(new Dosen23(kode, nama, jk, usia));  i++;  break;  case 2:  dosen.tampil();  break;  case 3:  dosen.sortingAsc();  System.out.println("Data setelah sorting ASC:");  dosen.tampil();  break;  case 4:  dosen.sortingDesc();  System.out.println("Data setelah sorting DSC:");  dosen.tampil();  break;  case 5:  System.out.println("Keluar dari program.");  return;  default:  System.out.println("Pilihan tidak valid!");  }  }  }    } |

2. Output

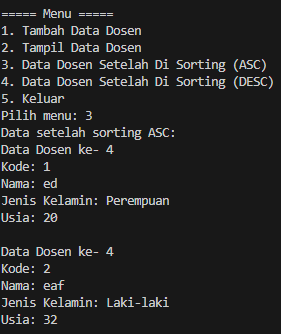
Tambah data



Tampil Data



Ascending



Descending

